"Applying Architecture Modeling Methodology to Enterprise Software Domains"

SSTC Conference
May16, 2011

Dr. Joey Rivera

Report Documentation Page

Form Approved
OMB No. 0704-0188

1. REPORT DATE	2. REPORT TYPE	3. DATES COVERED
16 MAY 2011		**00-00-2011 to 00-00-2011**

4. TITLE AND SUBTITLE	5a. CONTRACT NUMBER
Applying Architecture Modeling Methodology to Enterprise Software Domains	5b. GRANT NUMBER
	5c. PROGRAM ELEMENT NUMBER
6. AUTHOR(S)	5d. PROJECT NUMBER
	5e. TASK NUMBER
	5f. WORK UNIT NUMBER
7. PERFORMING ORGANIZATION NAME(S) AND ADDRESS(ES) **Rivera Group,555 Maple Ave,Sellersburg,IN**	8. PERFORMING ORGANIZATION REPORT NUMBER
9. SPONSORING/MONITORING AGENCY NAME(S) AND ADDRESS(ES)	10. SPONSOR/MONITOR'S ACRONYM(S)
	11. SPONSOR/MONITOR'S REPORT NUMBER(S)

12. DISTRIBUTION/AVAILABILITY STATEMENT
Approved for public release; distribution unlimited

13. SUPPLEMENTARY NOTES
Presented at the 23rd Systems and Software Technology Conference (SSTC), 16-19 May 2011, Salt Lake City, UT. Sponsored in part by the USAF. U.S. Government or Federal Rights License

14. ABSTRACT

15. SUBJECT TERMS

16. SECURITY CLASSIFICATION OF:			17. LIMITATION OF ABSTRACT	18. NUMBER OF PAGES	19a. NAME OF RESPONSIBLE PERSON
a. REPORT **unclassified**	b. ABSTRACT **unclassified**	c. THIS PAGE **unclassified**	**Same as Report (SAR)**	**25**	

About Dr. Rivera

- **President of Rivera Group**
- **Specialization**: Enterprise Software Development, Research & Development, Business Process Reengineering (BPR)
- **R&D Projects**: Modeling & Simulation, Natural Language Processing (NLP) of Open Source data for the Intelligence Community

- SBA 8(a) Small Disadvantaged Business
- Service-Disabled Veteran Owned Small Business
- HubZone Certified Minority-Owned Business

EAGLE6 MODELING

Journal Publications

Rivera, J., Auguston, M., Finkbine, R. (2011) Modeling Methodology for Validation and Verification of System Architecture Designs, 23rd Annual Systems & Software Technology Conference (SSTC 2011), Salt Lake City, Utah.

Rivera, J. (2010). Applying Architecture Modeling Methodology to the Naval Gunship Software Safety Domain. ACM/IEEE 13th International Conference on Model Driven Engineering Language and Systems. Oslo, Norway.

Rivera, J., & Luqi. (2010). Requirements Framework for the Software System Safety Technical Review Panel Technical Review Package. Monterey, CA: Naval Gunnery Project Office PEO WS3C.

Rivera, J., Luqi, and Berzins, V. (2009) Effective Programmatic Software Safety Strategy for US Navy Gun System Acquisition Programs, 6th Annual Acquisition Research Symposium of the Naval Postgraduate School:, Vol. 2, Monterey, CA.

Predict System Failures Before You Build

Question: "I have a great idea, but what happens if I change my system architecture?"

- **Eagle6** is a modeling and simulation tool that is capable of predicting system behavior.

Ensure Changes to System Architecture Do Not Cause Unintended System Behavior

Validates Architectural Designs That Mix Legacy Systems and New Technologies

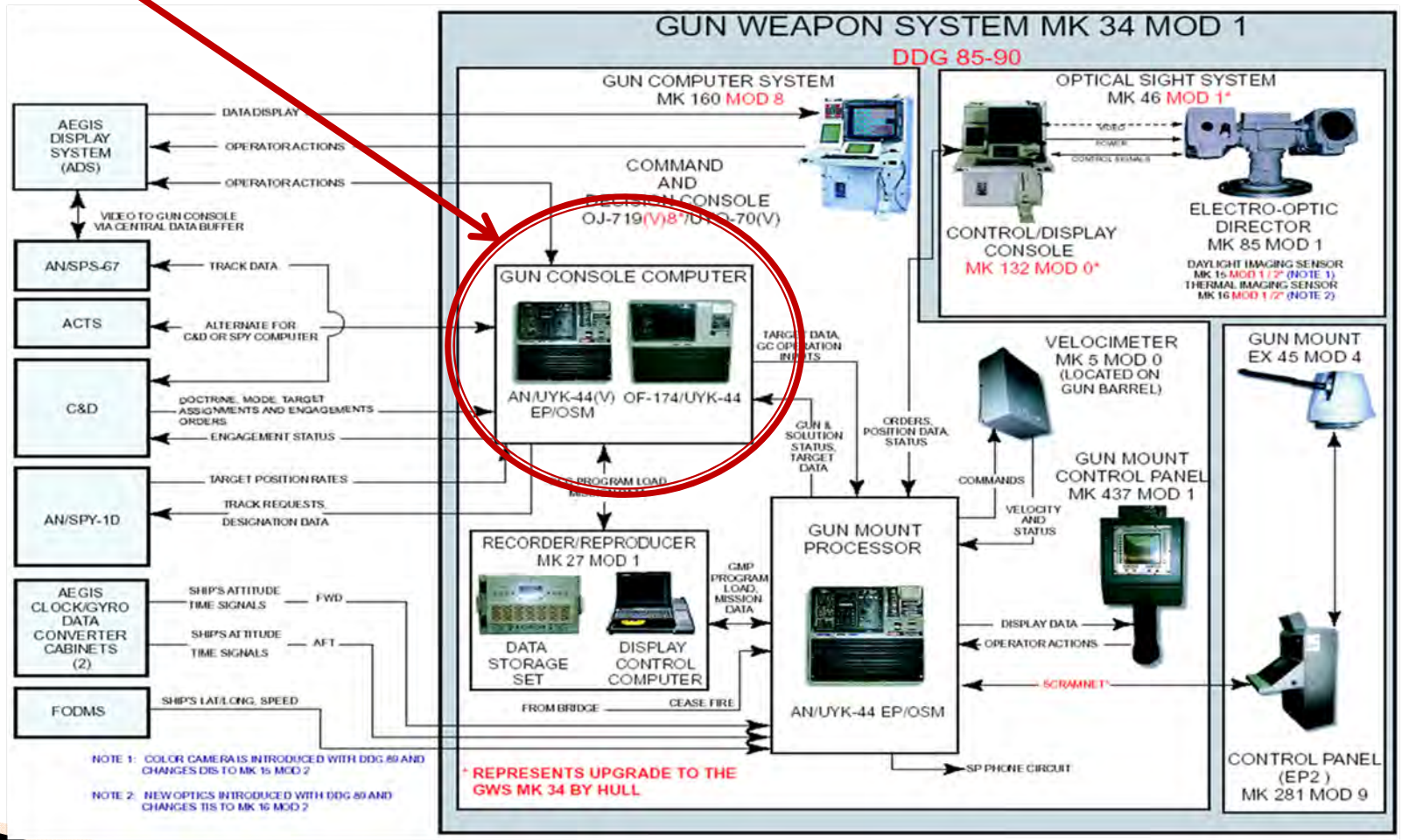
Evaluates Business Process Reengineering Designs (Lean Six Sigma)

Answer: Test **BEFORE** You Invest!

EAGLE6 MODELING

System of System Dependencies

Problem: What could happen if I upgrade/change this system?



Coming Up: How we solve this problem…

Problem Statement

"System Architecture Designs May Contain Unintended and/or Unknown System Behavior."

Issues with Requirements

Typical difficulties

- Users initial concept of system is nebulous
- Users description of system is incomplete and inconsistent
- Users (usually) don't understand what they really need
- Interpreting users description of problem is error-prone
- Perception of system changes during analysis, requires reworking
- Different users will view the system differently

Question: How do we know if the system architecture represents all system requirements?
Answer: Model Checking

Produce Better Software

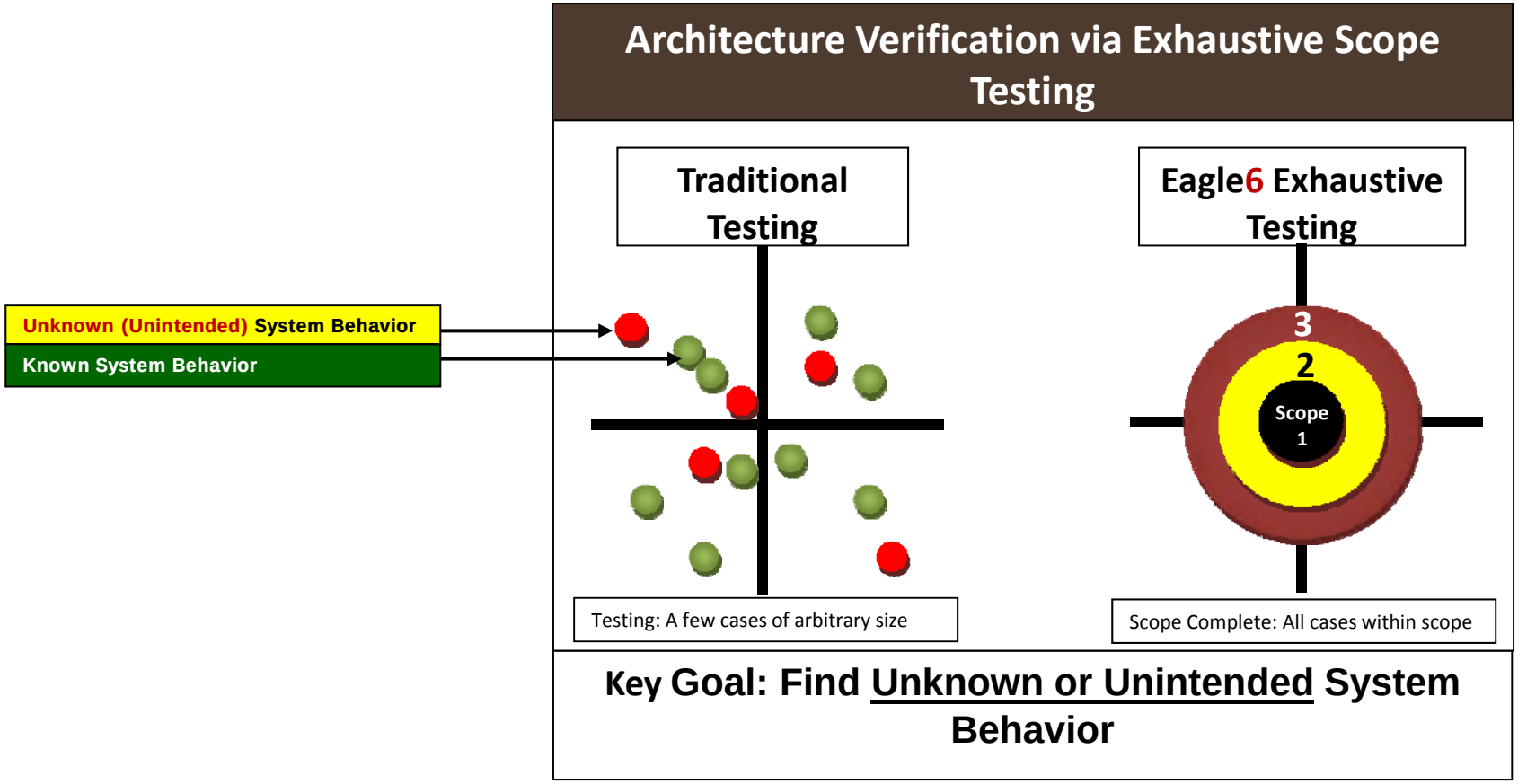
- Major goal of software engineers
  - Develop reliable systems
- Formal Methods
  - Mathematical languages, techniques and tools
  - Used to specify and verify systems
  - Goal: Help engineers construct more reliable systems
- A mean to examine the entire state space of a design (whether hardware or software)
  - Establish a correctness or safety property that is true for all possible inputs

Problems with Formal Methods

- Past years of the formal methods
  - Obscure notation
  - Non-scalable techniques
  - Inadequate tool support
  - Hard to use tools
  - Very few case studies
  - Not convincing for practitioners

- Bottom Line: It's not easy.

Unknown System Behavior



Model Checking

Pros

- Great for system safety testing.
  - Medical Systems
  - Weapon Systems
- Great for finding unknown system behavior and/or architectural design flaws (assertion checking)

Cons

- Modeling languages are very complex and require domain expertise
- Models require a very long time to develop
- Modifying models is not easy, making reuse very difficult

Test Using One Model

- Requirements Testing
- Conformance Testing
- Functional Testing
- Integration Testing
- Black Box Testing
- Performance Testing
- Regression Testing
- System Testing
- Unit Testing

Using a single model allows for system constraints to remain resident throughout all stages of the system lifecycle.

Summary of Formal Methods

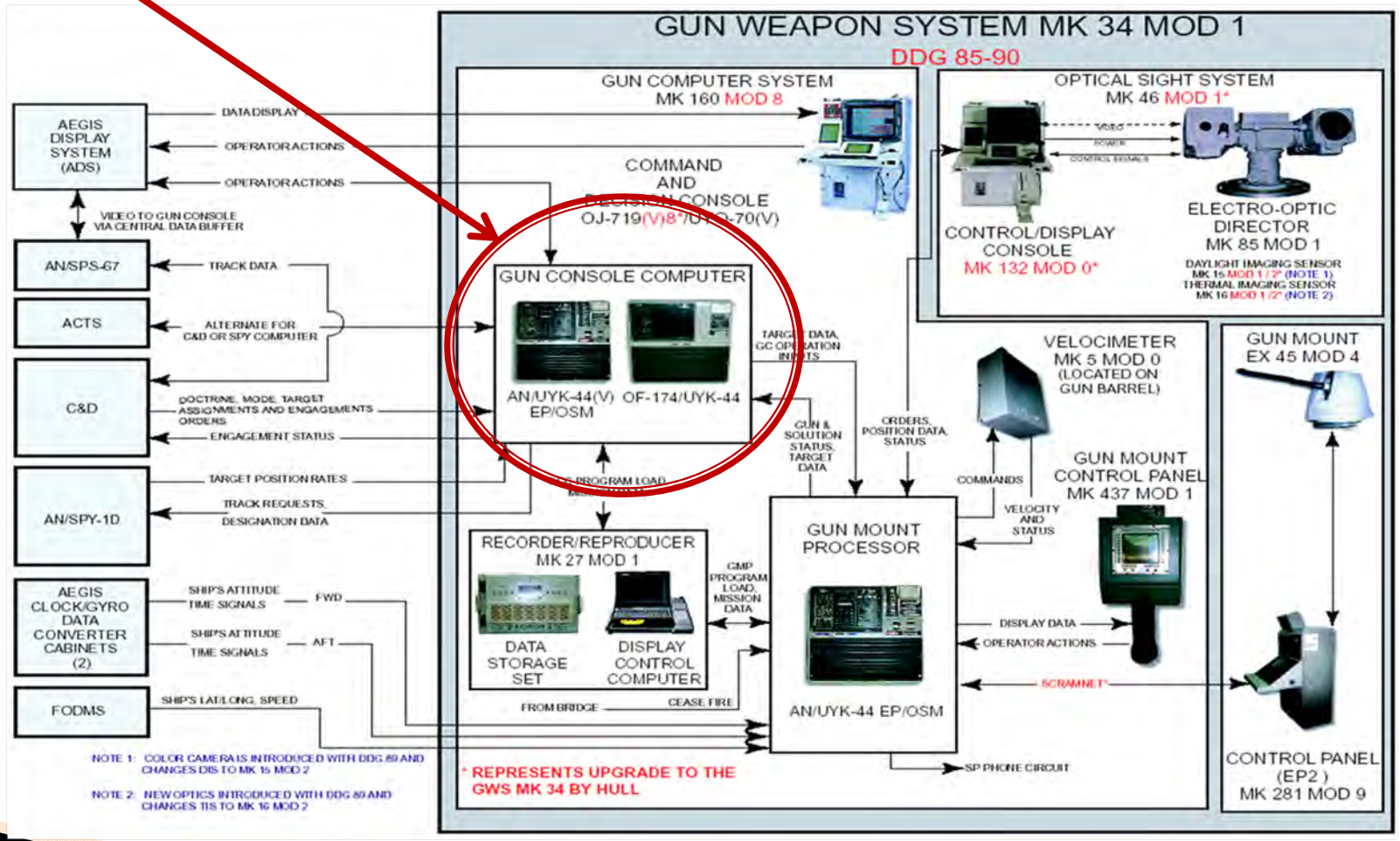
- Formal methods can be applied at various points through the development process
  - Specification
  - Verification
- **Specification**: Give a description of the system to be developed, and its properties
- **Verification**: Prove or disprove the correctness of a system with respect to the formal specification or property

Eagle6 Demonstration (www.Eagle6.com)

EAGLE6 MODELING

Old SoS Problem, New Solution

Problem: What could happen if I upgrade/change this system?



Answer: **Model the solution and test, test, test!**

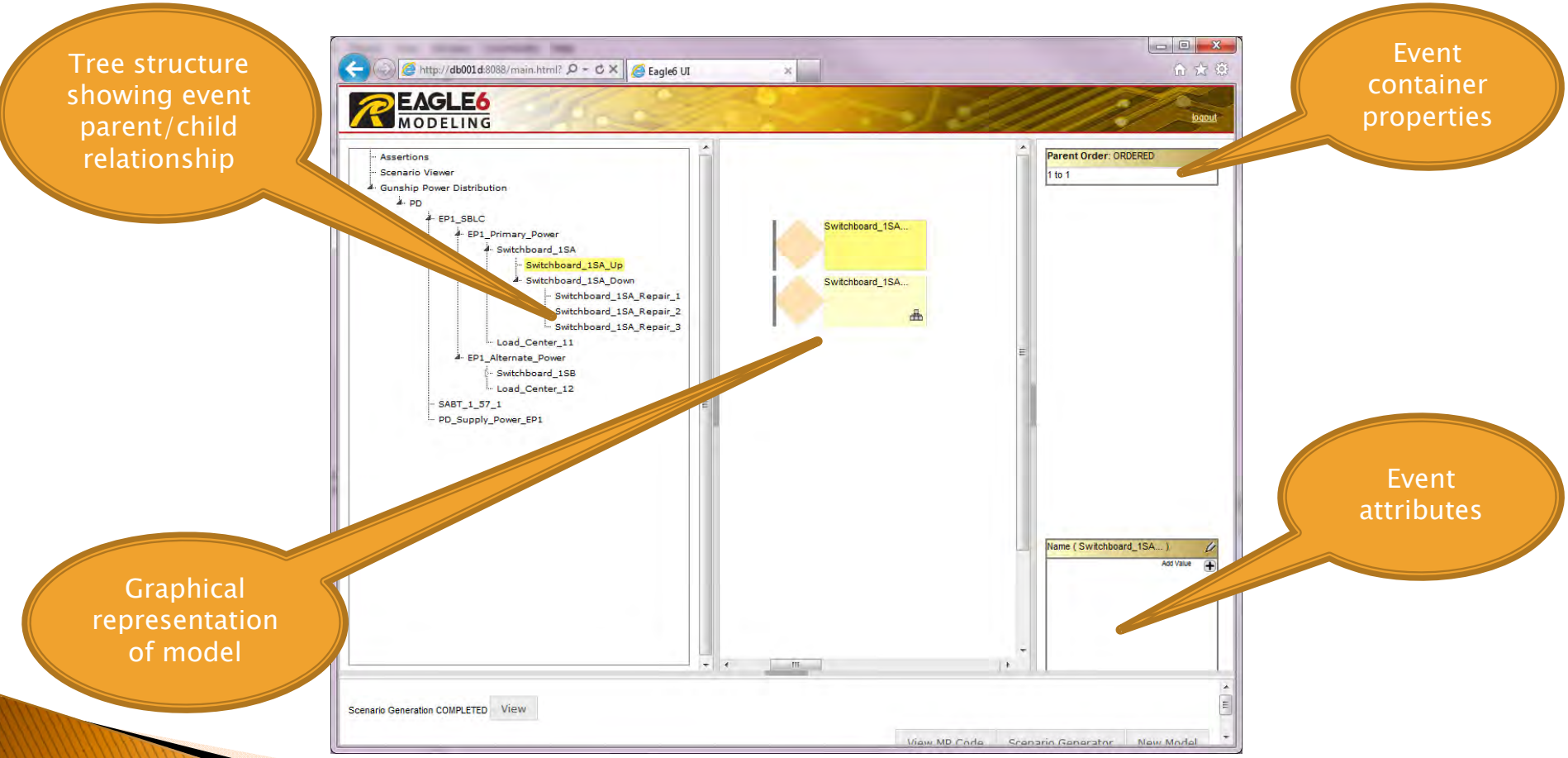
EAGLE6 MODELING

Reminder: Test Using One Model

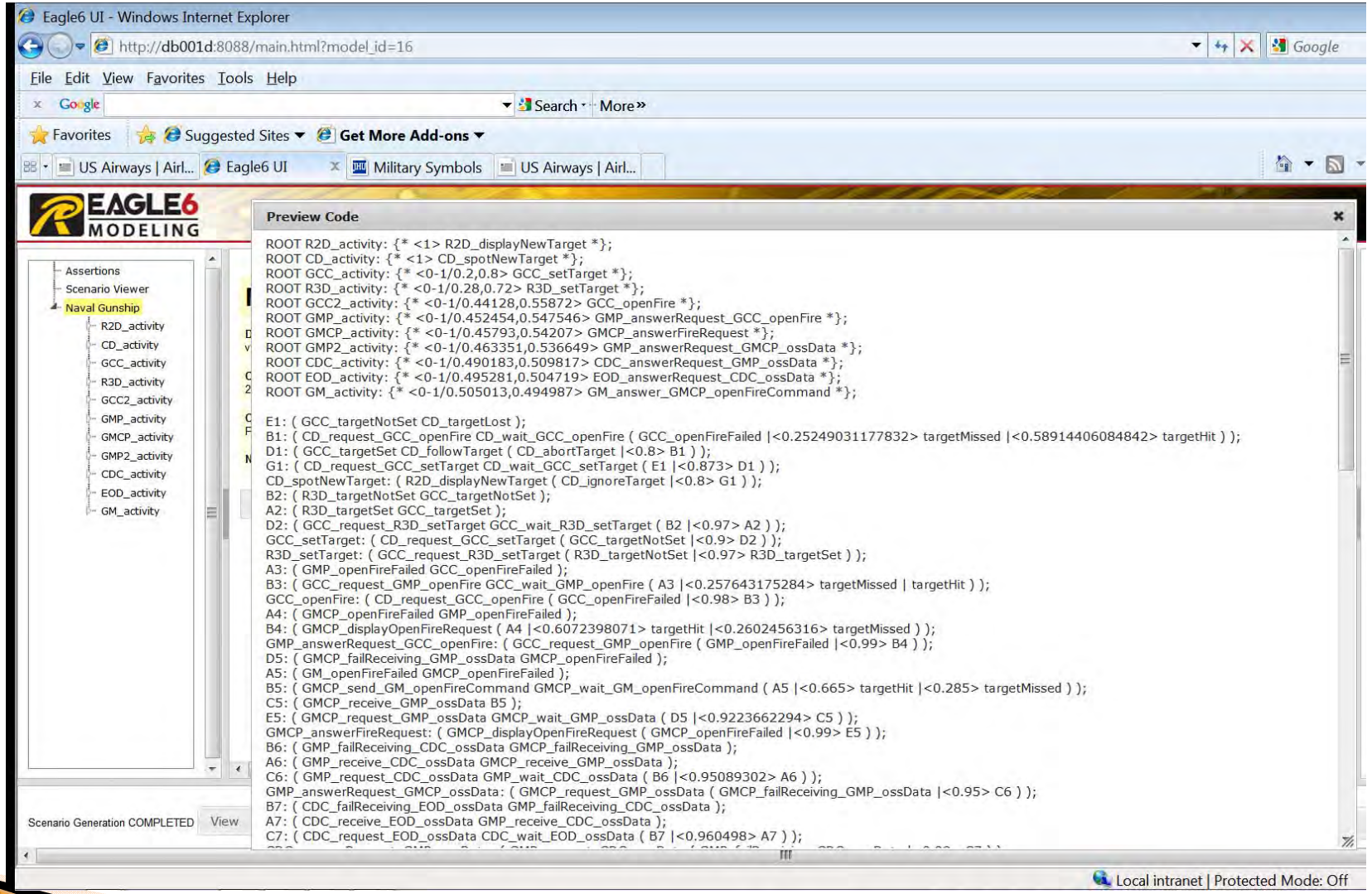
- Requirements Testing
- Conformance Testing
- Functional Testing
- Integration Testing
- Black Box Testing
- Performance Testing
- Regression Testing
- System Testing
- Unit Testing

Using a single model allows for system constraints to remain resident throughout all stages of the system lifecycle.

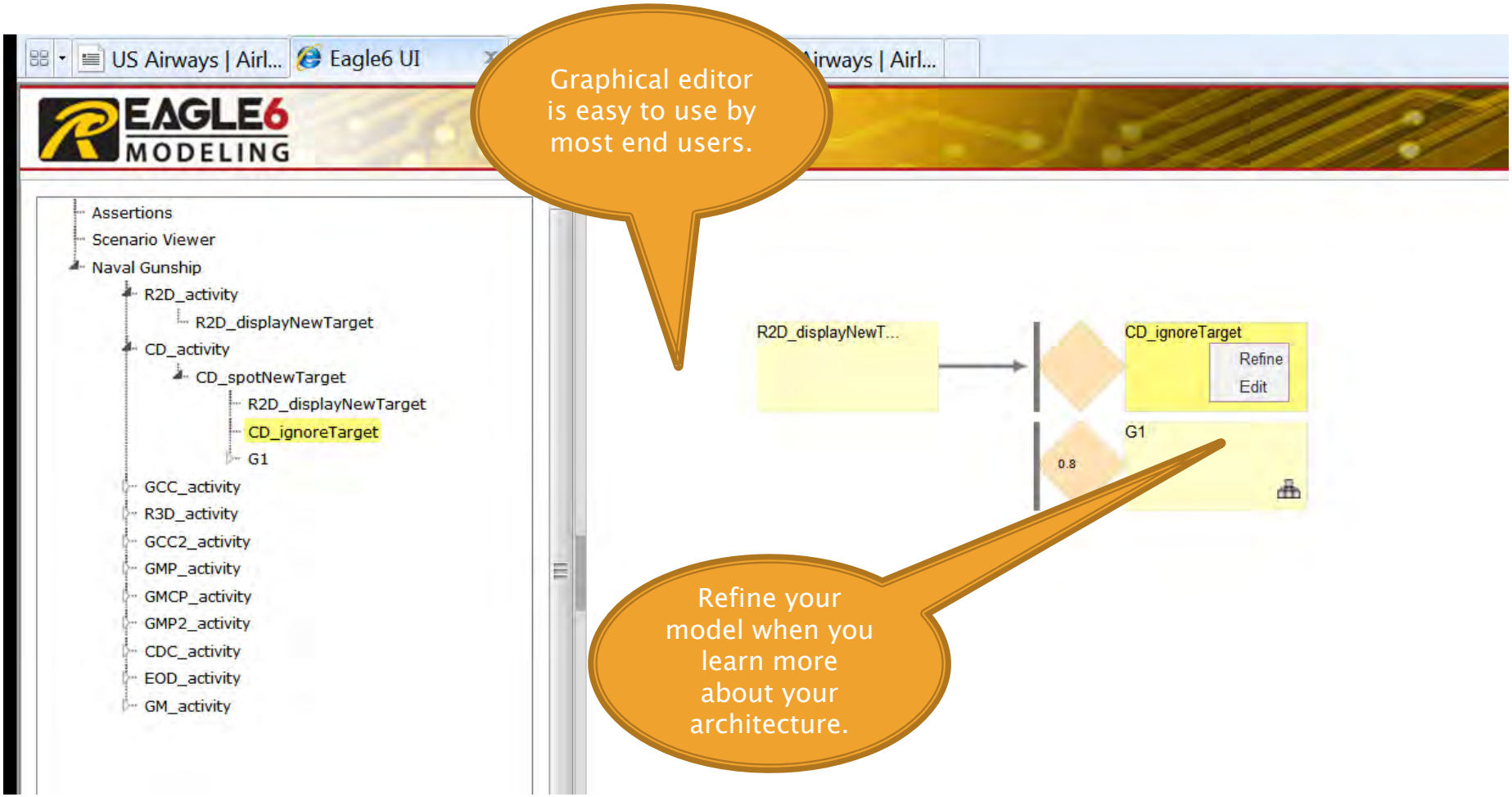
Model the Enterprise Without a Single Line of Code



Eagle6 Model in Text Form

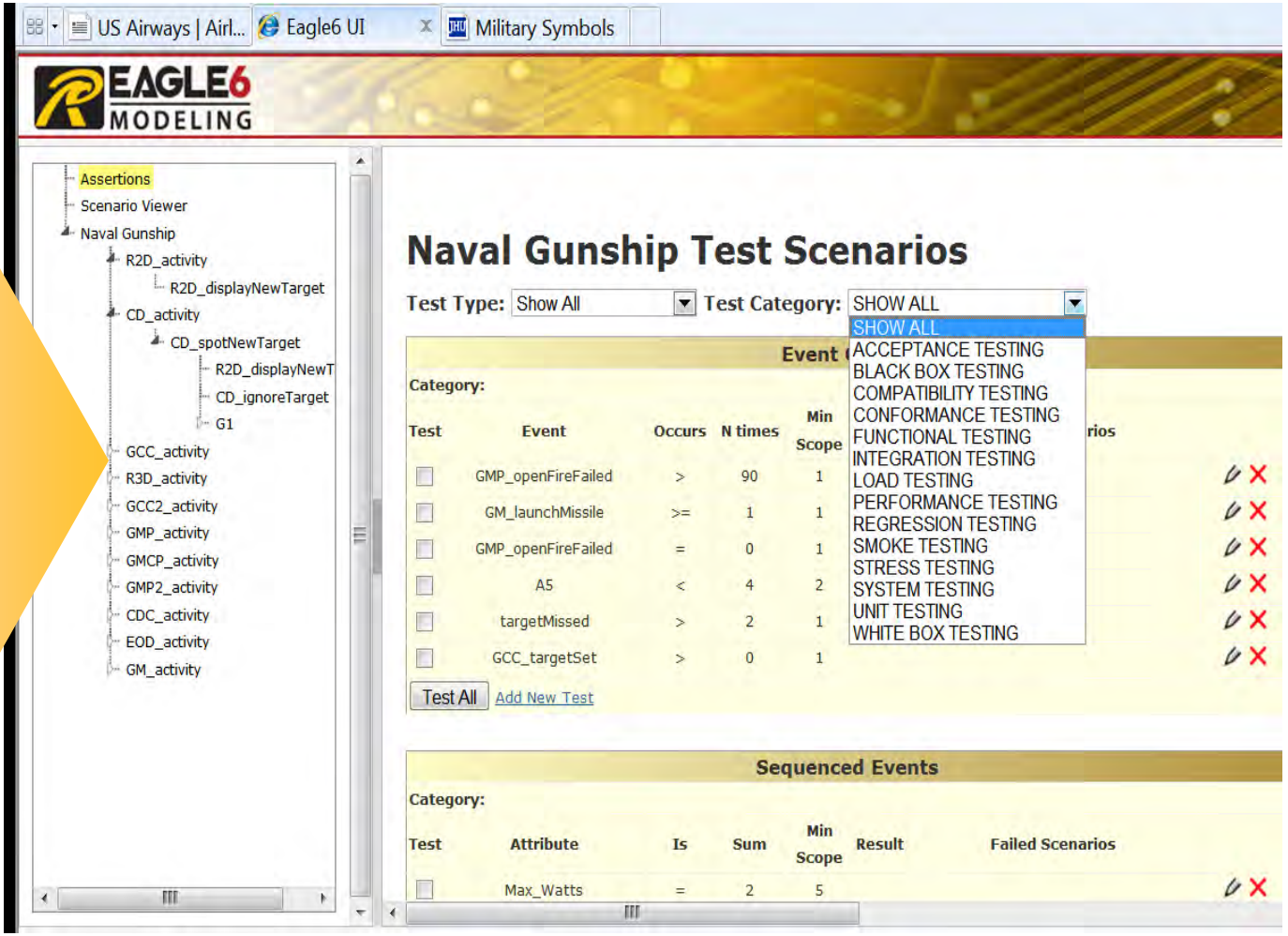


Eagle6 Graphical Editor



Test SoS Assertions

- Requirements Testing
- Conformance Testing
- Functional Testing
- Integration Testing
- Black Box Testing
- Performance Testing
- Regression Testing
- System Testing
- Unit Testing



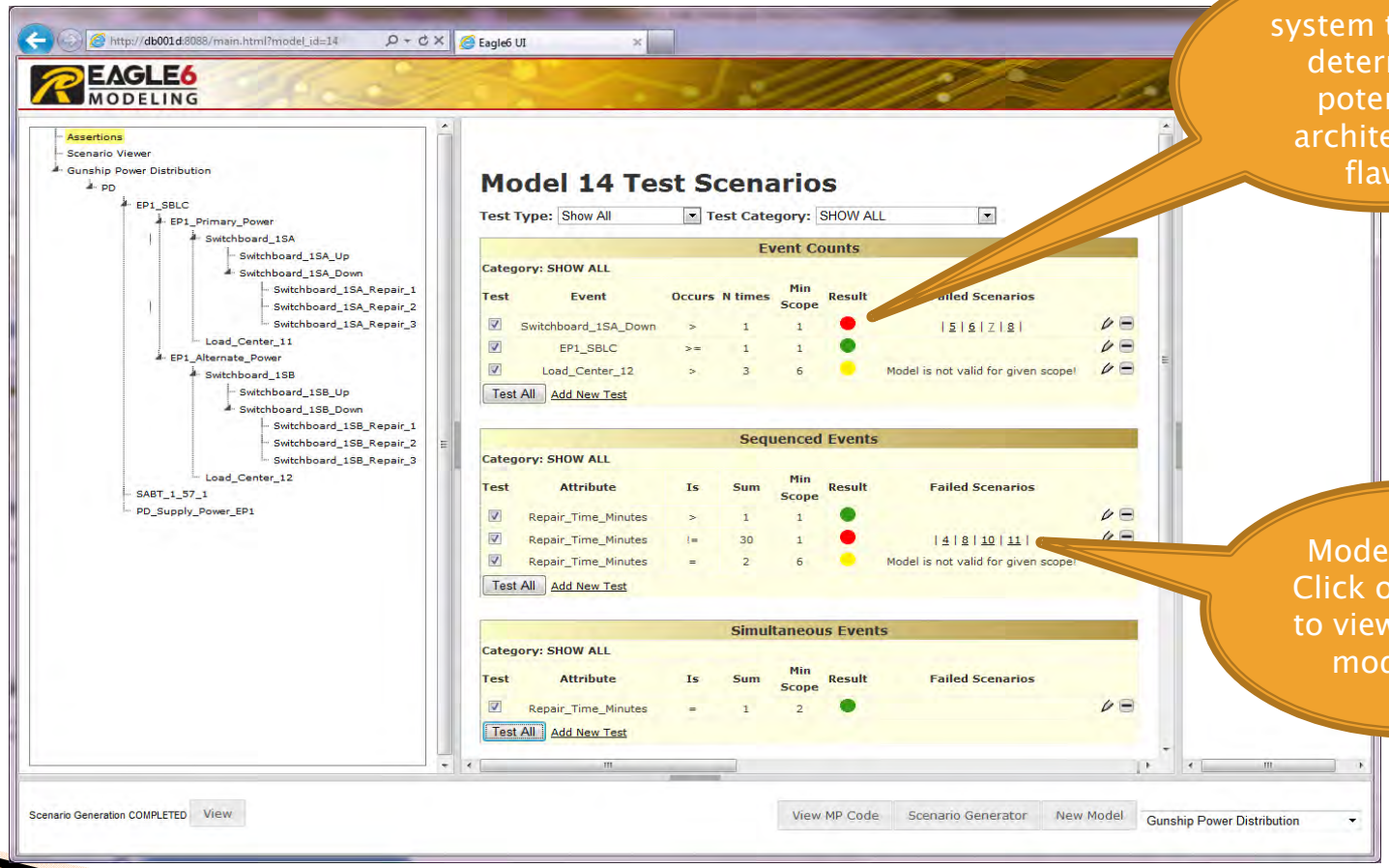
Key: Tests via Queries

1) **Event Count** – Check for the existence of a specific system state

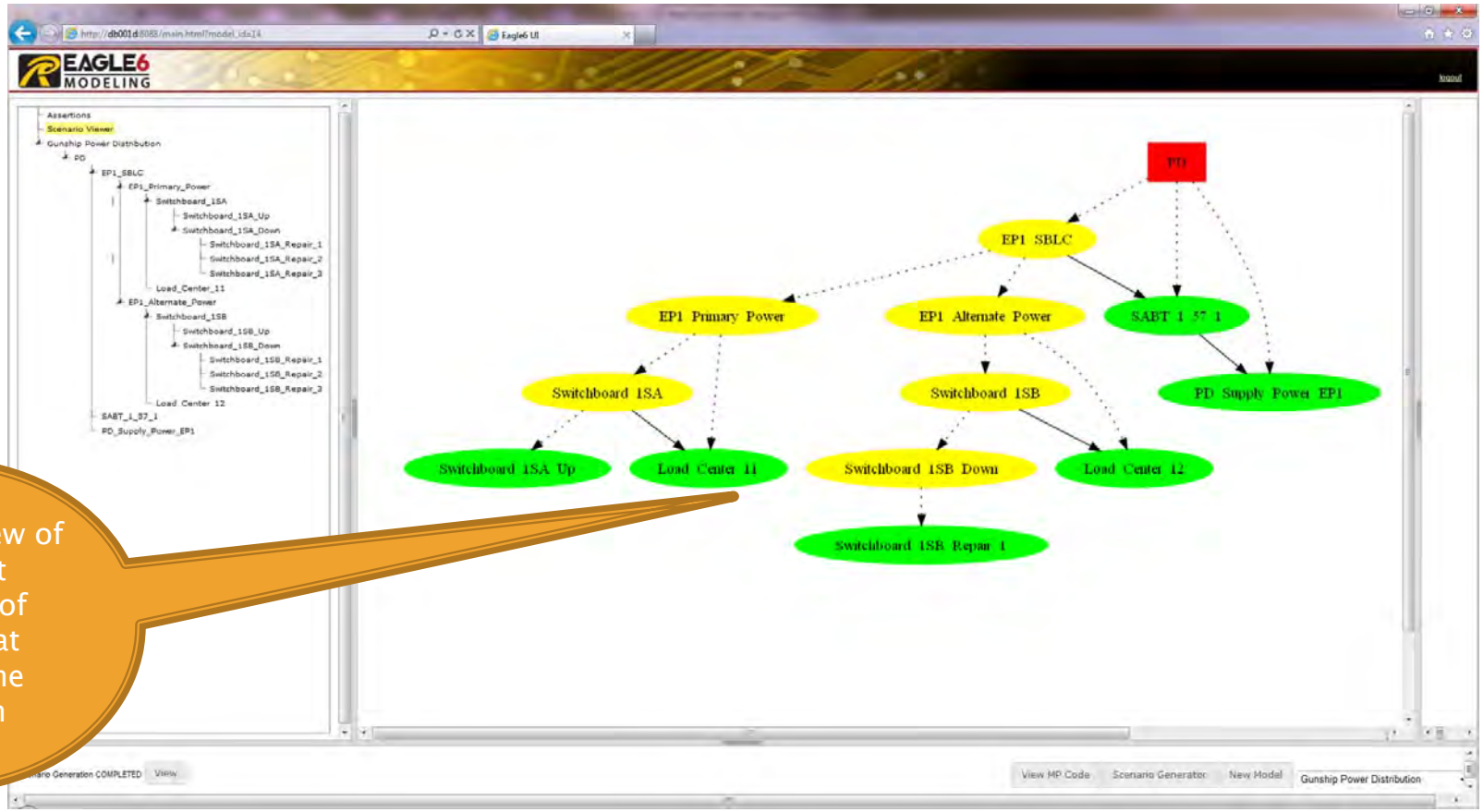
2) **Sequence of Events** – can a series of events happen?

3) **Simultaneous Events** – Can a combination of events happen?

Run All System Tests with One Model in Seconds



Scenario Viewer



Graphical view of the exact sequence of events that violated the assertion

Eagle6 Summary

1. Modeling and Simulation software tool that is used to dynamically model any type of complex enterprise system to identify **risks** in system architecture.

2. Checks all possible system states within the model scope.

3. Capable of executing all types of system tests within a single model.

4. Modeling interface that allows a user to write models without having to learn a modeling language (tool is designed for the average user)

RIVERA GROUP

Questions?

Dr. Joey Rivera
jrivera@riverainc.com

Phil Lushin
plushin@riverainc.com

812-246-4055

SBA 8(a) Certified SDB

SDVOSB Service Disabled Veteran Owned Small Business CVE

HUBZone Historically Underutilized Business Zone